Verifiable Poll-site e-Voting

Indisputable computer elections at polling stations

Cross-Reference To Related Application

This application claims the benefit of U.S. Provisional Patent Application No. 60/428,334, filed on November 21, 2002, and which is hereby incorporated by reference in its entirety.

Introduction

Trust in elections relies on two key principles: voter verification and election verification. This is easily illustrated with old-fashioned pencil-and-paper, hand-counted ballots.[1]

Voter Verification happens when you (the voter) visually check your voted ballot to ensure that your choices are reliably recorded on the ballot paper. You place your voted ballot in the ballot box in clear view of election "trustees" which typically includes election officials, public observers, members of the press, party officials, and voters. Voter Verification provides you with convincing evidence that your vote was recorded as cast without trusting the recording medium.[2]

When the polls close, the voted ballots are dumped out and hand-counted in plain view of these trustees. *Election Verification*[3] happens when the trustees are satisfied with the accuracy of the counting. If they are not, the ballots are recounted until all trustees are satisfied. Since *all* the ballots are kept in plain view of all trustees, it is straightforward to prove that

(1) no ballots have been added or deleted,
(2) no ballots have changed since cast, and
(3) anyone can reproduce the election results.

The simplest election is a "show of hands" election where you raise your hand to indicate your choice. It is Voter Verifiable because you know whether your hand is up or not. It is Election Verifiable because all trustees (which includes voters for a "show of hands" election) can count each hand raised. "Show of hands" elections, however, are not private and don't scale very well for large elections.

Machine counted paper voting systems are Voter Verifiable because you can check that your vote is recorded on paper as you intended. But they are not Election Verifiable because no one can prove that (1) no ballots were added or delete, (2) ballots were not changed since cast, and (3) all ballots were counted to product the final election results. The US Presidential Election in 2000 showed that machine counted paper ballots could be lost or inadvertently changed after casting and it took a decision of the US Supreme Court to decide the election.[4]

Simple DRE (direct recording electronics) voting systems are neither Voter Verifiable nor Election Verifiable. You cannot check that your ballot accurately represents your choices and there is only *procedural* assurance that all ballots have accurately been counted to produce the election results.

To engender trust, election systems should be both Voter Verifiable and Election Verifiable. Figure 1 shows the verification properties for common election methods.

---

[1] In this case, hand-counted ballots are meant to include those elections where ballots are relatively simple and easily counted by hand.

[2] For example, Voter Verification protects the recording of your ballot from interference by malicious computer software (like viruses, Trojan horses, and worms) whether that malicious software was introduced by outside hackers, election insiders, or even the election manufacturer.

[3] More formally, public *Election Verification* ensures that any party can check unambiguously that the election is fair, which means that the published final tally is consistent with the correctly cast ballots. Any party can perform this check by inspecting a public transcript of the election, consisting of all public information (including voted ballots) and all the messages exchanged before, during and after the election. This inspection must also meet accepted voter privacy requirements.

[4] TBD press reference to punch cards changing after fed through counters.

	Voter Verifiable	Election Verifiable
"Show of hands"	Yes	Yes
Hand-counted paper	Yes	Yes
Machine counted paper	Yes	No
DRE	No	No

Figure 1 *Verification properties of common election methods.*

Voting and verifying your vote at the polling station

Upon entering the polling station, the poll-worker checks you in, typically by having you sign your name on the poll book.[5] The poll-worker gives you a *Voting Token* (like a smart card or key) that determines the appropriate ballot for your precinct.

Armed with your Voting Token, you select a voting machine, which is a computer with or without a touchscreen. You insert your Voting Token into the voting machine and your vote *Verification Dictionary* is printed on a receipt printer next to the voting machine. The Verification Dictionary allows you to later verify your vote, which will be discussed in a moment.[6] The Verification Dictionary, shown in Figure 2, shows every possible choice for a given race and a unique *Verification Code* corresponding to each choice.

To ensure that the voting machine is not programmed to cheat, it is important for the Verification Dictionary to be printed before you vote. Then, the voting machine must *commit* to the Verification Code corresponding to a given candidate. Since the election trustees assemble the Verification Dictionaries before the election, an independent election auditor can check the Verification Dictionary at anytime during the voting day to make sure that the voting machine is not printing erroneous Verification Dictionaries. This checking procedure is explained later.

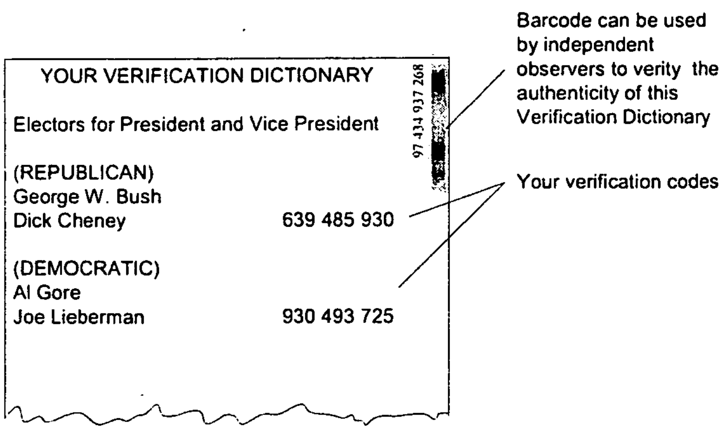


Figure 2 *Your verification dictionary*

To make your selections, all candidates for a given race are listed on a single screen and include an area for submitting a write-in candidate. If desired by the jurisdiction, "none-of-the-above" and "instant runoff" voting can also be supported. Once you make a selection, a checkmark appears on the screen and the background changes color. After selecting a candidate, you press the *Next* button to continue. The final screen shows a summary of all your selections including those races that you skipped. You press *Change* to change any of your selections.

Once satisfied with your selections, you press *Verify Ballot* to verify your vote. At this point a cash-register printer prints your *Vote Receipt* with verification codes corresponding to your selections as in Figure 3. For your selections, if the codes on your receipt match (see Figure 4) those on your Verification Dictionary, press *Cast*

[5] Voter authentication policies vary widely from mere voter declaration to requirements for picture identification.

[6] This description assumes all voters are given the opportunity to verify their vote. Vote verification could also be an option provided to voters that are interested in verifying that their vote was counted as cast. In this case, you would be given an option to either print your Verification Dictionary or view you ballot.

Ballot to make your vote official. If they don't match, retry or contact a poll-worker for assistance. Once your ballot is cast, keep your receipt to later verify that your ballot is counted in the final tally.

Once your ballot is cast, you must destroy your Verification Dictionary. Destruction of your Verification Dictionary, which should be enforced by a poll-worker, protects your privacy, protects you from coercion, and will not anyone to buy your vote.

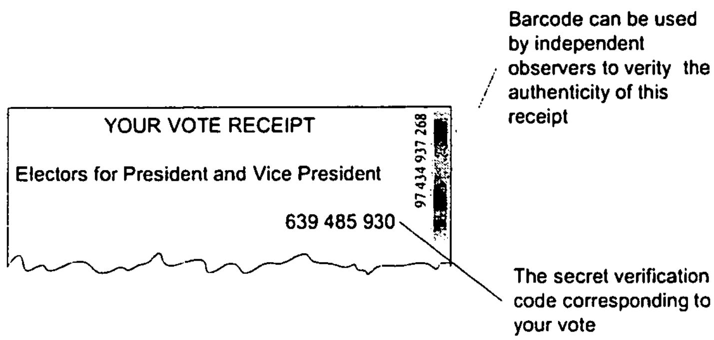


Figure 3 *Your vote receipt*

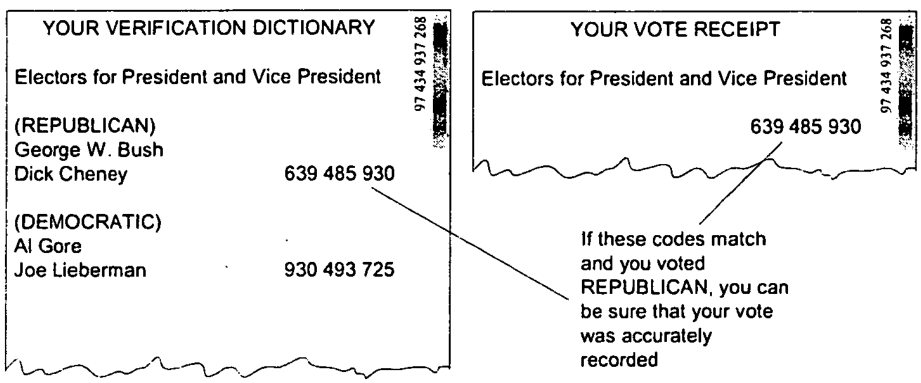


Figure 4 *Verifying your vote*

A variation that may make the code comparisons easier is to visually compare them. The same voting procedure is followed except that the Verification Dictionary and Voter Receipt codes are printed on transparent sheets. You lay your Vote Receipt over your Verification Dictionary and a checkmark appears next to your selected choice (Figure 5). Details of this will be explained later.

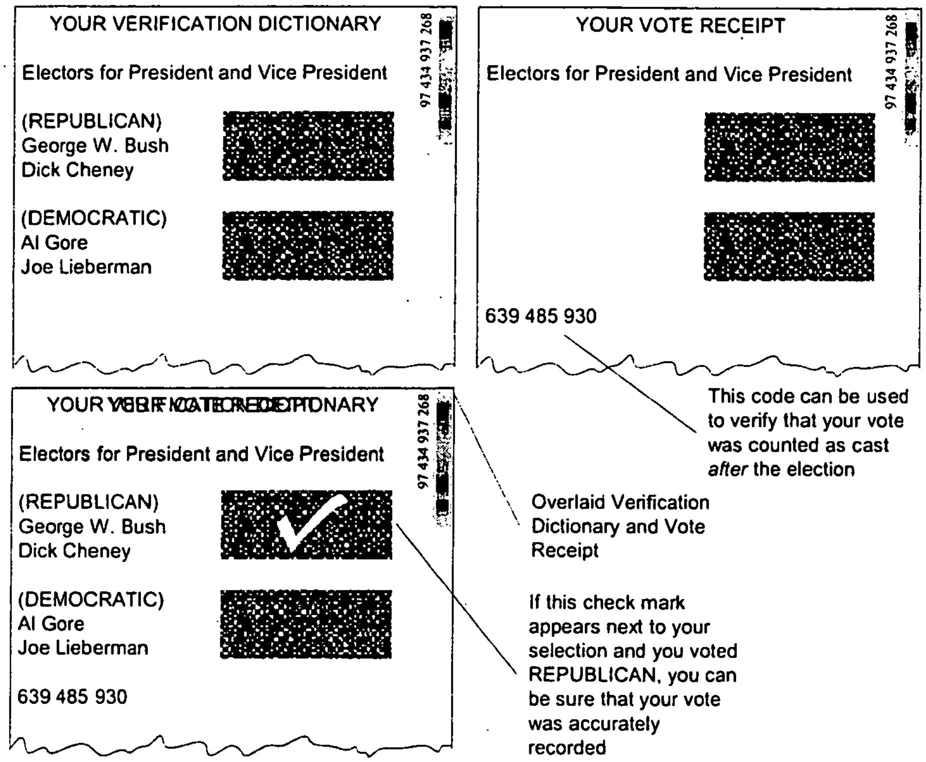


Figure 5 Verifying your vote visually

Verifying your vote after the election

After the polls close and the *Election Transcript* is published, you can use your Vote Receipt to check that the vote you cast in the polling station is the same one that was counted in the official election results. Your Vote Receipt number is published online for each race along with the Verification Code corresponding to your selection. If the verification on your receipt matches that on the published online receipt, you can be sure that your vote was counted as cast (Figure 6). Since you do not have your Verification Dictionary (it was destroyed at the polling station), you cannot prove to others how you voted.

BEST AVAILABLE COPY

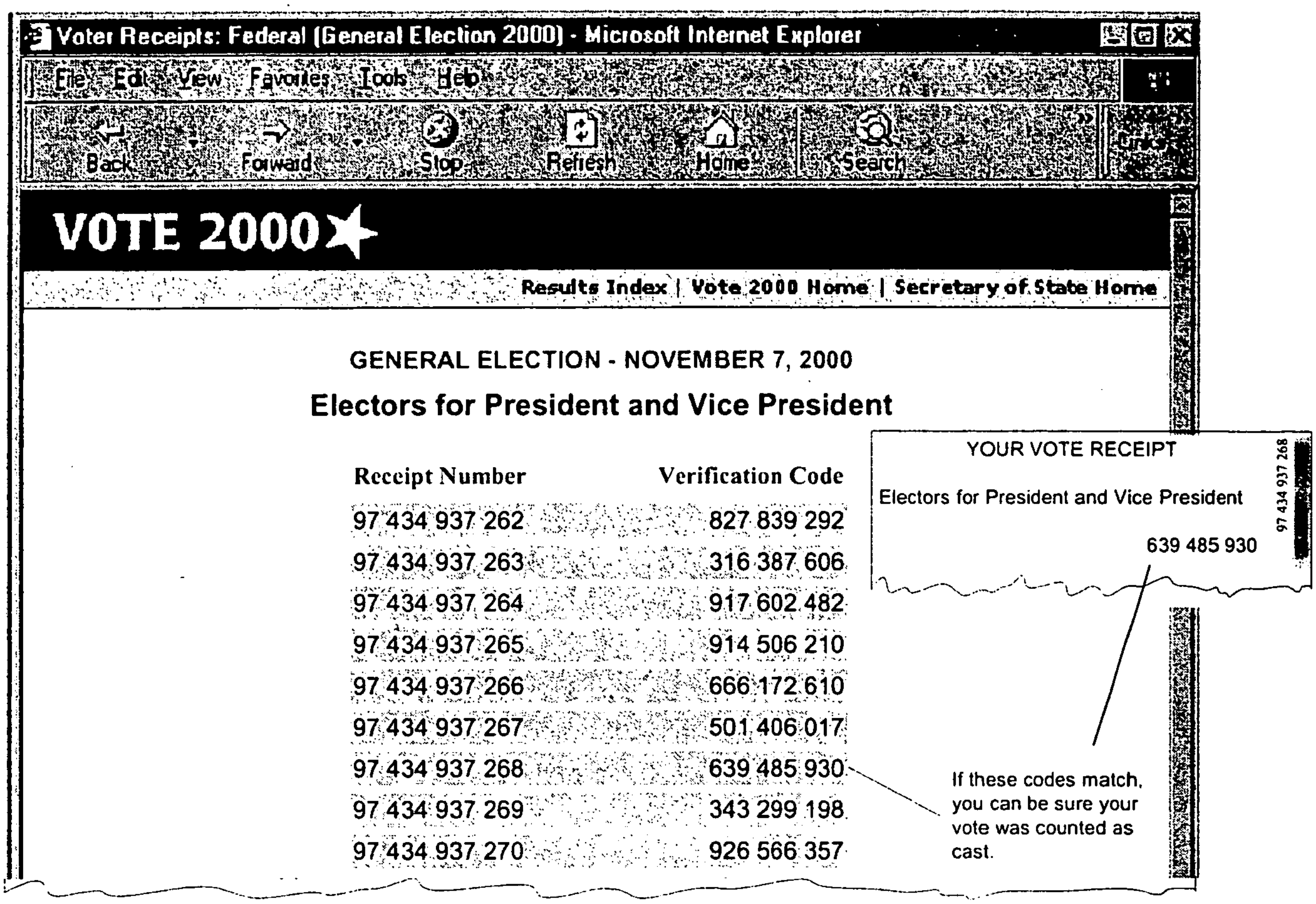


Figure 6 Verifying your vote online after the election

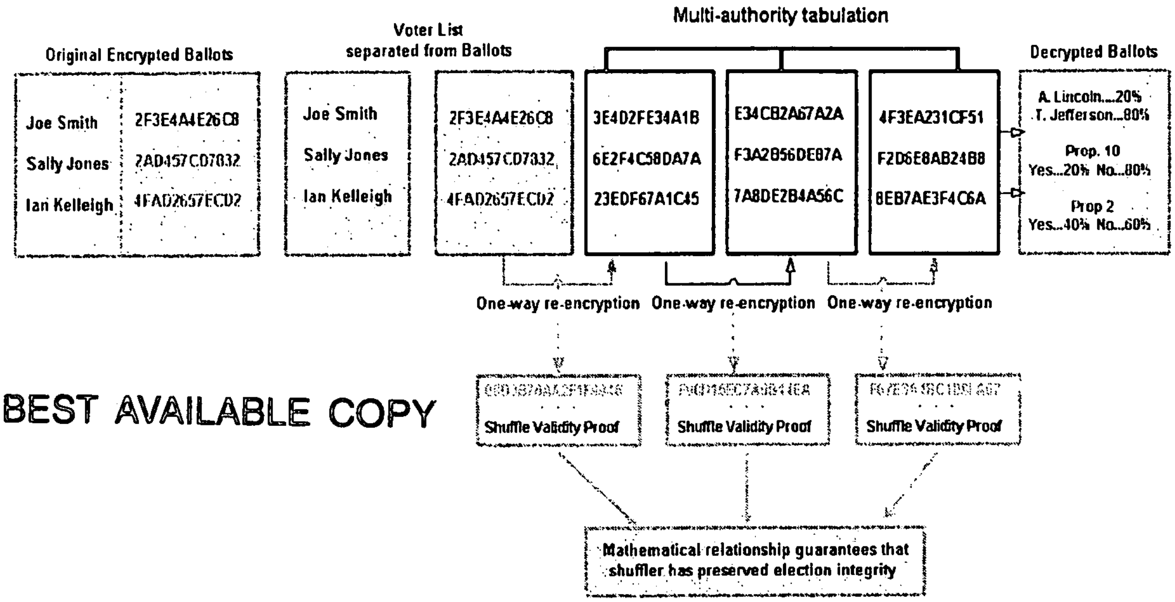
Verifying the entire election

Once the polls close, the voted ballots are "shuffled" by each trustee to separate each voter's identity from their vote. This shuffle also produces a *validity proof* that proves that no ballots were added, deleted, or changed during the shuffling (Figure 7).

Once shuffled, the trustees decrypt and tabulate the ballots to produce the final election results. Only if all the trustees collude can voter privacy be violated or the election be defrauded. This argues for adding more trustees that are at cross-purposes. This is the same distributed trust configuration that protects hand-counted paper elections.

The encrypted ballots, shuffled ballots, and validity proofs are then posted online in what is called an *Election Transcript*. At this point, anyone can download the election transcript and execute an open source program to complete the *Election Verification*.

BEST AVAILABLE COPY



BEST AVAILABLE COPY

Figure 7 *Shuffling and tabulating voted ballots*

Security properties

- You can be sure that your vote was recorded accurately.
- You can be sure that your vote was included in the final election tally.
- If there is inside cheating, there is physical evidence of such cheating.
- The trustees protect election integrity, and all must collude to corrupt the election results.
- The trustees protect your secret ballot, and all must collude to violate your privacy.

<u>How Anyone can Audit Verification Dictionaries</u>

Why it works

<u>How Visual Comparison Works</u>

More formally

Much of the setting for the conventional voting application can be found in [CGS97]. Votes are submitted as ElGamal pairs (or a sequence of these pairs if more data is required), where the voter's choices are encoded in some standard form.

Once the polls are closed, the independent trustees sequentially shuffle the ballots. On output of the final shuffle, the final collection of encrypted ballots is decrypted in accordance with the threshold scheme, and the clear text votes are tabulated in full view by normal election rules.

The trustees, who participate in the sequential shuffles, may be arbitrary in number, and they may be completely different from those who hold shares of the election private key. The sequence of ballots that are finally decrypted can only be matched with the original sequence of submitted ballots if all of the shuffling authorities collude, since each of their permutations is completely arbitrary.

Each trustee performs a shuffle on all ballots, in turn, as follows:

1. A number is chosen secretly, randomly and independently.
2. Each vote is re-encrypted and replaced.
3. A Chaum-Pedersen proof is published without revealing the secrets. A shuffle with secret exponent c is performed on the resulting encrypted votes.
4. Steps 1 and 2 are repeated.

Background

As early as 1992, Fujioka et al [FOO92] applied David Chaum's blind signature primitive cryptographic function [Cha81] to elections. Actually, Chaum suggested the election application as early as 1981. This scheme was later found to only support *Voter Verification*—that is, only individual voters could verify that their vote was received and counted correctly; an independent body could not verify that the entire election was free from fraud. This property, public *Election Verification* (also known as *Universal Verifiability*), was introduced in 1997 [CGS97]. However, it is also evident in election schemes introduced by Benaloh as early as 1986 [BY86].

Since it is difficult to get citizens to vote, let alone ensuring that all voters verify, *Voter Verification* is not sufficient to protect an entire election. Public *Election Verification* is sufficient to prove election validity even in the face of insider fraud. However, *Voter Verification* can (1) protect client-side malicious code (e.g., viruses, Trojan horses, worms, etc.) from corrupting voted ballots and (2) provide voters convincing evidence that their individual vote was counted as cast.

Blind signature e-voting systems, like [FOO92], represent the first generation of e-voting technology. They do not represent the leading e-voting candidates. In fact, the property of public *Election Verification*, should be applied to all e-voting systems, whether online or offline. Essentially, *Election Verification* ensures that any party can check unambiguously that the election is fair, which means, among other things, that the published final tally is consistent with the correctly cast ballots. Any party can perform this check by inspecting a public transcript of the election, consisting of all public information (including voted ballots) and all the messages exchanged before, during and after the election. This inspection must also meet accepted voter privacy requirements.

Conclusions

Acknowledgements

References

[BY86] J. Benaloh, M. Yung. *Distributing the power of a government to enhance the privacy of voters.* ACM Symposium on Principles of Distributed Computing, pp. 52-62, 1986.

[CGS97] R. Cramer, R. Gennaro, B. Schoenmakers. *A secure and optimally efficient multi-authority election scheme.* Advances in Cryptology - EUROCRYPT '97, Lecture Notes in Computer Science, Springer-Verlag, 1997.

[Cha81] D. Chaum. *Untraceable electronic mail, return addresses, and digital pseudonyms.* Communications of the ACM, 24(2): 84-88, 1981.

[FOO92] A. Fujioka, T. Okamoto, K. Ohta. *A practical secret voting scheme for large scale elections.* Advances in Cryptology–AUSCRYPT '92, Lecture Notes in Computer Science, pp. 244-251, Springer-Verlag, 1992.

[Nef01] C. A. Neff. *Verifiable, Secret Shuffles of ElGamal Encrypted Data for Secure Multi-Authority Elections.* Eighth ACM Conference on Computer and Communications Security (CCS-8), November 2001.

The above references are incorporated herein by reference, as are the following U.S. applications bearing serial numbers:

09/534,835
09/534,836
09/535,927
09/816,869
09/989,989
10/038,752
10/081,863
10/367,631
60/445,502